Aplikasi Kriptografi File Menggunakan Metode Blowfish dan Metode Base64 pada Dinas Kependudukan dan Pencatatan Sipil Kota Tangerang Selatan

Mujito
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
jitosalemba@gmail.com

Anugrah Bagus Susilo
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
anugrahbagussusilo@gmail.com

***Abstrak*- Dinas kependudukan Dan Pencatatan Sipil Kota Tangerang Selatan adalah suatu instansi pemerintah yang menangani masalah kependudukan dan catatan sipil warga daerah Tangerang data-data pencatatan sipil meliputi data akte pernikahan, data akte kelahiran dan akte kematian. Data – data tersebut merupakan data yang bersifat rahasia jika data tersebut diketahui oleh orang yang tidak bertanggung jawab dapat disalah gunakan seperti data kartu tanda penduduk, jika jatuh ke tangan orang yang salah dan digunakan untuk melakukan tindakan kriminal seperti peminjaman uang dan lain sebagainya. Untuk mengatasi data tersebut diperlukan aplikasi yang dapat menyandikan atau meng*enkripsi* data tersebut, Dimana *mekanisme*nya nanti data-data tersebut discan terlebih dahulu kemudian *file image* hasil *scan* akan di *enkripsi* menggunakan algoritma *Blowfish* terlebih dahulu baru kemudian di enkripsi lagi dengan algoritma *Base64* , aplikasi menggunakan bahasa pemprograman PHP. Data dapat diamankan dengan *kriptografi* algortima *Blowfish* dan *Base64*. Data tidak dapat dibuka oleh pihak yang tidak berhak yang tidak memiliki kunci untuk *enkripsi* dan *dekripsi file*.**

Kata kunci-- Kriptografi, Blowfish, Base64, enkripsi, dekripsi

I. PENDAHULUAN

Dinas kependudukan Dan Pencatatan Sipil Kota Tangerang Selatan adalah suatu instansi pemerintah yang menangani masalah kependudukan dan catatan sipil warga daerah Tangerang data-data pencatatan sipil meliputi data akte pernikahan, data akte kelahiran dan akte kematian. Data – data tersebut merupakan data yang bersifat rahasia jika data tersebut diketahui oleh orang yang tidak bertanggung jawab dapat disalah gunakan seperti data kartu tanda penduduk, jika jatuh ke tangan orang yang salah dan digunakan untuk melakukan tindakan kriminal seperti peminjaman uang dan lain sebagainya. Dengan banyaknya data-data penting yang sering juga bersifat rahasia, maka pihak Dinas Kependudukan dan Pencatatan Sipil Kota Tangerang Selatan ingin membuat supaya data tersebut tidak dapat dibaca secara langsung, karena data tersebut di sandikan atau dienkripsi baru kemudian disimpan ke dalam basis data, dan jika ingin melihat data tersebut menjadi data asli maka perlu di dekrpsi yaitu mengubah data yang terenkripsi menjadi data asli.

Untuk lebih mengamankan proses enkripsi dan deskripsi, perlu dilakukan suatu mekanisme yang memberikan sedikit kemungkinan agar data asli tidak bisa dibongkar oleh penyerang. Sehingga penulis menggunakan 2 metode dalam melakukan proses enkripsi dan deskripsi, yaitu *Blowfish* dan *Base64*. Dengan menggunakan kombinasi antara kunci algoritma *Blowfish* dan *Base64* diharapkan akan membuat pengamanan data memiliki tingkat keamanan yang lebih tinggi. Khususnya untuk data kependudukan dan catatan sipil sehingga data asli tersebut tidak dapat dibaca dan diterjemahkan oleh orang yang tidak bertanggung jawab.

Berdasarkan permasalahan tersebut penulis mengambil penelitian dengan judul *"*Aplikasi Kriptografi File Menggunakan Metode Blowfish Dan Metode Base64 Pada Dinas Kependudukan Dan Pencatatan Sipil Kota Tangerang Selatan*"*.

A. Batasan Masalah

Sesuai dengan judul skripsi ini, penulis memfokuskan masalah yang ada dan agar tidak menyimpang dari pokok bahasan maka penulis membuat batasan permasalahan yaitu:

- Pembuatan aplikasi menggunakan bahasa pemprograman PHP.

- Proses *enkripsi* dan *dekripsi* hanya dilakukan untuk data kependudukan dan pencatatan sipil yaitu berupa *file* berbentuk image (.jpeg).
- Maksimal *File* yang dienkripsi sebesar 2 Mb

II. LANDASAN TEORI

A. *Konsep Dasar Kriptografi*

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika dikirim dari suatu tempat ke tempat yang lain. (Ariyus, 2008) Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak lain yang tidak berhak untuk mengetahui isi pesan tersebut. Kriptografi diperlukan untuk menghindari pihak yang tidak berhak mengetahui isi dari pesan yang dikirimkan tersebut. Dengan adanya kriptografi, isi dari pesan akan diacak sedemikian rupa menggunakan algoritma kriptografi tertentu sehingga akan menghasilkan sebuah pesan yang acak yang tidak dapat dibaca sebelum isi pesan yang sebenarnya kembali dimunculkan menggunakan algoritma kriptografi tersebut. (Schneier, 1996)

B. *Enkripsi*

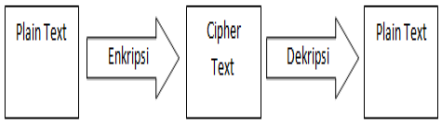
Merupakan hal yang sangat penting dalam *kriptografi*, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *plain text*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode. Sama halnya dengan jika tidak mengerti akan sebuah kata maka yang dilakukan adalah dengan melihatnya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah teks asli ke bentuk teks-kode digunakan algoritma yang dapat mengkodekan data yang diinginkan

C. *Dekripsi*

Merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (teks-asli), disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda atau kebalikan dengan algoritma yang digunakan untuk enkripsi

D. *Kunci*

Kunci yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*). Maka pesan atau data asli sebelum dienkripsi disebut *plain text*. Sedangkan pesan yang sudah diacak disebut *cipher text*. Proses pengubahan *plain text* menjadi *cipher text* disebut dengan enkripsi, sedangkan proses pengubahan *cipher text* kembali menjadi *plain text* disebut dengan dekripsi.



Gambar 1. Proses *Enkripsi-Dekripsi*

E. *Algoritma Blowfish*

Merupakan algoritma simetri yang tergolong dalam metode *block cipher*. Ada dua tipe dasar algoritma simetris yaitu *block cipher* dan *stream cipher*. Sebuah *block cipher* memproses *block byte* (biasanya 64 atau 128 *bit*) pada satu waktu. Sebuah *stream cipher* memproses satu *byte* atau bahkan satu *bit* pada suatu waktu. (Thorsteinson and Ganesh, 2003)

Blowfish dibuat oleh seorang *Cryptanalyst* bernama Bruce Schneier, yang merupakan Presiden perusahaan *Counterpane Internet Security.Inc*, dan dipublikasikan tahun 1994. Algoritma ini digunakan pada komputer yang mempunyai *microprocessor* besar (32-*bit* keatas dengan *cache* data yang besar). (Schneier, 1996)

Karateristik *Blowfish* adalah sebagai berikut:

- Merupakan *block cipher* dengan 64 *bit block*
- Panjang kunci merupakan *variable* dengan panjang kunci hingga 448 *bit*
- Mengenkripsi data pada *microprocessor* 32 *bit* dengan rata – rata 18 *clock cycle* per *byte*, lebih cepat dari DES dan IDEA.
- Tidak mempunyai hak paten dengan harga yang gratis.
- Dapat berjalan pada memori kurang dari 5 KB.
- Mempunyai struktur yang sederhana dan implementasi yang mudah. (Pachghare, 2009)

Blowfish terdiri atas dua bagian :

1) *Key-Expansion*
   Berfungsi merubah kunci (Minimum 32-*bit*, Maksimum 448-*bit*) menjadi beberapa *array* subkunci (*subkey*) dengan total 4.168 *byte*.
2) Enkripsi Data
   Terdiri dari iterasi fungsi sederhana (*Feistel Network*) sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi kunci *dependent* dan substitusi

kunci dan data *dependent*. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32-*bit* . Operasi tambahan lainnya hanyalah empat penelusuran tabel (*table lookup*) *array* berindeks untuk setiap putaran. (Schneier, 1996)

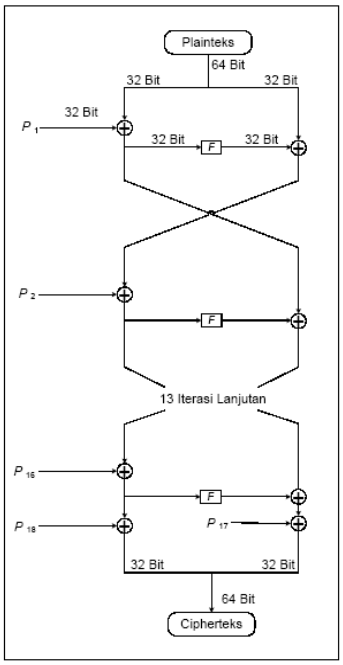
Blowfish menggunakan subkunci yang besar. Kunci tersebut harus dihitung sebelum enkripsi atau dekripsi data.

- Bentuk inisial *P-array* sebanyak 18 buah yaitu (P1,P2,……..P18) masing-msing bernilai 32-*bit* . *P-array* terdiri dari delapan belas kunci 32-*bit* subkunci: P1,P2,…….,P18
- Bentuk *S-Box* sebanyak 4 buah masing-masing bernilai 32-*bit* yang memiliki masukan 256. Empat 32-*bit* *S-Box* masing-masing mempunyai 256 masukan: (S1,0),(S1,1),……..,(S1,255) (S2,0),(S2,1),……..,(S2,255) (S3,0),(S3,1),……..,(S3,255) (S4,0),(S4,1),……..,(S4,255)

Blowfish adalah sebuah jaringan Feistel yang terdiri dari 16 putaran. *Input*-annya adalah elemen data 64 *bit*. Cara untuk melakukan enkripsi adalah sebagai berikut:

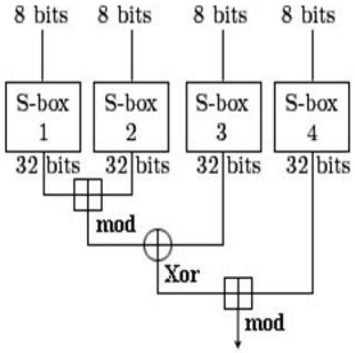
- Pertama-tama *plain text* yang akan dienkripsi diasumsikan sebagai masukan, *plain text* tersebut diambil sebanyak 64-*bit*, dan apabila kurang dari 64-*bit* maka tambahkan *bit* nya, supaya dalam operasi nanti sesuai dengan datanya.
- Hasil pengambilan tadi dibagi 2, 32-*bit* pertama disebut XL, 32-*bit* yang kedua disebut XR.
- Selanjutnya lakukan operasi berikut:
  For i = 1 to 16;
  XL = XL XOR Pi
  XR = F(XL) XOR XR
  Tukar XL dan XR
- Setelah iterasi ke-16, tukar XL dan XR lagi untuk melakukan *undo* pertukaran terakhir. Lalu lakukan
  XR = XR XOR P17
  XL = XL XOR P1

Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-*bit* kembali. (Scheiner, 1996).



Gambar 2. Jaringan Feistel Algoritma *Blowfish* (Scheiner, 1996)

Fungsi F adalah sebagai berikut:
Bagi XL menjadi empat bagian 8-*bit* :
a,b,c dan d. $F(XL) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) \text{ XOR } S_{3,c}) + S_{4,d} \bmod 2^{32}$.



Gambar 3. Fungsi F Pada Algoritma *Blowfish* (Scheiner, 1996)

Cara untuk melakukan dekripsi sama dengan cara untuk melakukan enkripsi seperti diatas, namun pada proses dekripsi urutan P1, P2, …, P18 digunakan dalam urutan terbalik.

Subkunci dihitung menggunakan algoritma *Blowfish*, metodanya adalah sebagai berikut:

- Pertama-tama inisialisasi *P-array* dan kemudian empat *S-Box* secara berurutan dengan *string* yang tetap. *String* ini terdiri *digit* hexadesimal dari p.
- *XOR* P_1 dengan 32 *bit* pertama kunci, *XOR* P_2 dengan 32 *bit* kedua dari kunci dan seterusnya untuk setiap *bit* dari kunci (sampai P_{18}). Ulangi terhadap *bit* kunci sampai seluruh P-*array* di *XOR* dengan *bit* kunci.

- Enkripsi semua *string* nol dengan algoritma *Blowfish* dengan menggunakan subkunci seperti dijelaskan pada langkah (1) dan (2).
- Ganti P_1 dan P_2 dengan keluaran dari langkah (3).
- Enkripsi keluaran dari langkah (3) dengan algoritma *Blowfish* dengan subkunci yang sudah dimodifikasi.
- Ganti P_3 dan P_4 dengan keluaran dari langkah (5).
- Lanjutkan proses tersebut, ganti seluruh elemen dari *P-array*, dan kemudian seluruh keempat *S-Box* berurutan, dengan keluaran yang berubah secara berlanjut dari algoritma *Blowfish.* (Scheiner, 1996)

F. Algoritma Base64

Transformasi *Base64* merupakan salah satu algoritma untuk *Encoding* dan *Decoding* suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan *encoding* (penyandian) terhadap data *extension*. Karakter yang dihasilkan pada transformasi *Base64* ini terdiri dari A..Z, a..z dan 0..9, serta ditambah dengan dua karakter terakhir yang bersimbol yaitu + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data *extension* atau istilahnya disebut sebagai pengisi pad. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. Dalam *Encoding Base64* dapat dikelompokkan dan dibedakan menjadi beberapa kriteria yang tertera

Tabel 1. Tabel indeks *Base64*

Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64
0	A	16	Q	33	h	50	y
1	B	17	R	34	i	51	z
2	C	18	S	35	j	52	0
3	D	19	T	36	k	53	1
4	E	20	U	37	l	54	2
5	F	21	V	38	m	55	3
6	G	22	W	39	n	56	4
7	H	23	X	40	o	57	5
8	I	24	Y	41	p	58	6
9	J	25	Z	42	q	59	7
10	K	26	a	43	r	60	8
11	L	27	b	44	s	61	9
12	M	28	c	45	t	62	+
13	N	29	d	46	u	63	/
14	O	30	e	47	v	pad	=
15	P	31	f	48	w		
16	Q	32	g	49	x		

Teknik *encoding Base64* sebenarnya sederhana, jika ada satu (*string*) *bytes* yang akan disandikan ke *Base64* maka caranya adalah sebagai berikut:

Misal kita ingin menyandikan teks MAN

- Ubah huruf – huruf yang akan di encripsi menjadi kode – kode ASCII

Text content	M	a	n
ASCII	77	97	110

- Kode – kode ASCII tersebut ubah lagi menjadi kode Biner

Text content	M	a	n
ASCII	77	97	110
Bit pattern	0 1 0 0 1 1 0 1	0 1 1 0 0 0 0 1	0 1 1 0 1 1 1 0

- Bagi kode biner tersebut menjadi hanya 6 angka per blok dan berjumlah kelipatan 4 blok
- Jika angka biner tidak berjumlah 6 angka dan 4 blok maka akan di tambah kode biner 0 sehingga mencukupi menjadi 4 blok
- Blok – blok tsb ubah kembali menjadi kode desimal (data di baca sebagai index)

Text content	M	a	n
ASCII	77	97	110
Bit pattern	0 1 0 0 1 1 0 1	0 1 1 0 0 0 0 1	0 1 1 0 1 1 1 0
Index	19	22	5	46

- Hasil kode index tersebut di ubah menjadi huruf yang ada pd index

Text content	M	a	n
ASCII	77	97	110
Bit pattern	0 1 0 0 1 1 0 1	0 1 1 0 0 0 0 1	0 1 1 0 1 1 1 0
Index	19	22	5	46
Base64-Encoded	T	W	F	u

- Jika nilai blok adalah hasil tambahan (0) maka hasil dari index tersebut bernilai '='

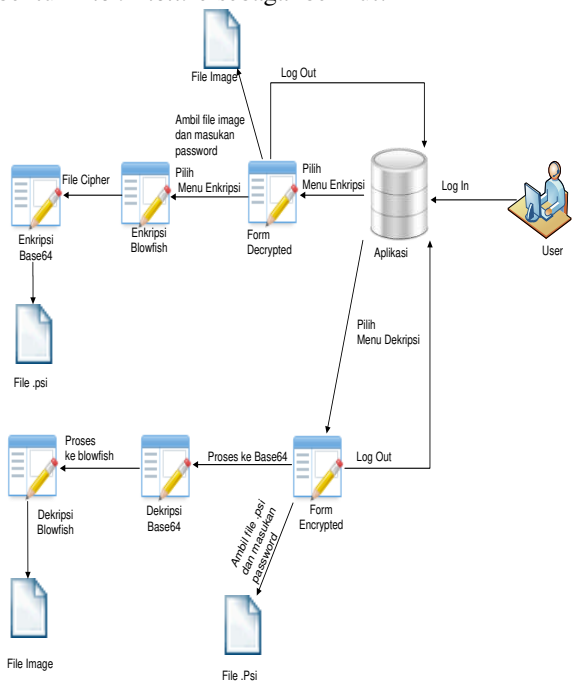
Text content	M			
ASCII	77			
Bit pattern	0 1 0 0 1 1 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Index	19	16	(kosong)	(kosong)
Base64-Encoded	T	Q	=	=

III. ANALISA MASALAH DAN RANCANGAN SISTEM

A. Analisa Masalah

Pengamanan data pada suatu instansi pemerintah merupakan hal yang sangat penting dan vital mengingat data tersebut tidak boleh diketahui oleh masyarakat umum data pada dinas kependudukan dan pencatatan sipil seperti data catatan sipil yaitu data kematian, data kelahiran dan data pernikahan. Untuk mengatasi data tersebut agar tidak diketahui oleh masyarakat umum maka diperlukan suatu aplikasi yang dapat menyandikan

atau meng*enkripsi* data tersebut, penulis ingin membuat aplikasi pengamanan data pada Dinas Kependudukan dan Pencatatan Sipil menggunakan Algoritma Blowfish dan Base64. Dimana mekanismenya nanti data-data tersebut discan terlebih dahulu kemudian *file image* hasil *scan*an akan di enkripsi menggunakan algoritma blowfish terlebih dahulu baru kemudian di enkripsi lagi dengan algoritma Base64 kemudian baru disimpan kedalam basis data, jika kita ingin melihat data hasil enkripsi menjadi data asli maka kita mendekrip data tersebut. Jika digambarkan dalam bentuk *Rich Picture* sebagai berikut:



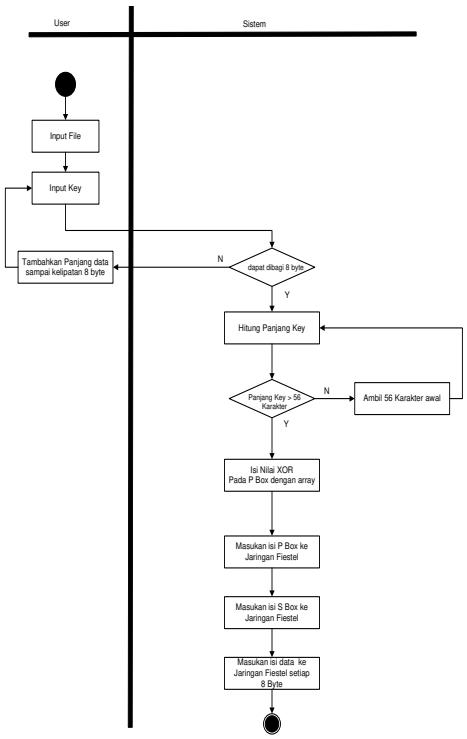
Gambar 4. *Rich Picture* Aplikasi

B. Activity Diagram

Activity Diagram digunakan untuk menjelaskan proses kegiatan yang terjadi pada program.

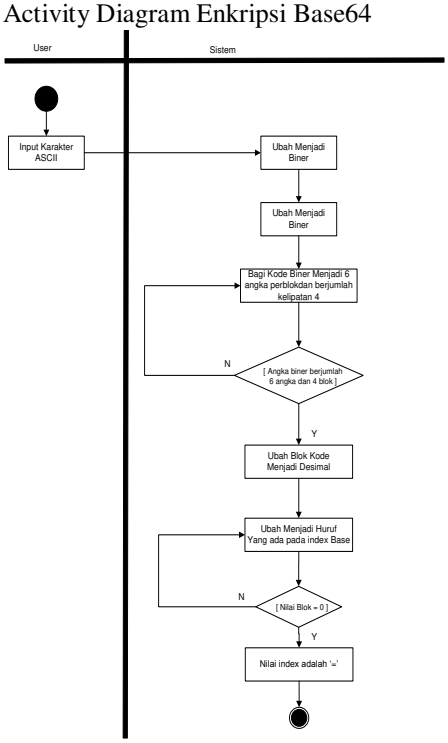
Berikut adalah *Activity Diagram* pada proses enkripsi *Blowfish:*

1) Activity Diagram Enkripsi Blowfish



Gambar 5. Tampilan *Activity Diagram Blowfish*

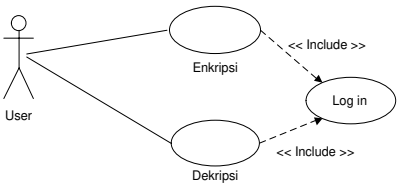
2) Activity Diagram Enkripsi Base64



Gambar 6. Tampilan *Activity Diagram Base64*

C. Use Case Diagram

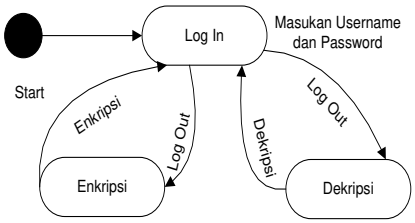
Use case diagram menjelaskan bagaimana seorang user dalam menjalankan aplikasi *enkrpsi* dan *dekripsi*



Gambar 7. *Use Case Diagram*

D. State Diagram

State diagram digunakan untuk men*deskripsi*kan perilaku sistem. *State diagram* men*deskripsi*kan semua kondisi yang mungkin muncul sebagai sebuah *object* begitu pula dengan *event*.



Gambar 8. *State Diagram*

Program start kemudian Login dan akan tampil pilihan menu Enkripsi dan *Dekripsi* dan pada menu *enkripsi* dan *dekripsi* terdapat fungsi *log out*

IV. IMPLEMENTASI DAN ANALISA PROGRAM

Dalam melakukan uji coba untuk mengetahui hasil dari proses enkripsi dan dekripsi tersebut, maka kebutuhan komputer yang harus dipenuhi adalah sebagai berikut.

A. Tampilan Layar Form Log in

User yang akan menggunakan aplikasi enkripsi harus log in terlebih dahulu.

Gambar 9. Tampilan Layar Log In

B. Tampilan Layar Form Enkripsi

Gambar 10. Tampilan Layar uji coba program halaman *Encrypter*

V. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan serta uji coba sistem dapat disimpulkan sebagai berikut :

- Data dapat diamankan dengan kriptografi algortima *Blowfish* dan *Base64.*
- Data tidak dapat dibuka oleh pihak yang tidak berhak yang tidak memiliki kunci untuk *enkripsi* dan *dekripsi file.*
- *File* yang diamankan menggunakan aplikasi ini tidak dapat dibuka oleh aplikasi lain.
- Program sistem keamanan dengan sistem *kriptografi* algoritma *Blowfish* dan *Base64* telah diuji coba, sehingga program dinyatakan sudah sesuai.

B. Saran

Pengembangan yang perlu dilakukan untuk penelitian berikutnya adalah sebagai berikut:

- Aplikasi ini mengenkripsi dan mendekripsi *file image* dan beberapa file menggunakan algoritma *Blowfish* dan *base64* untuk itu bisa di gunakan dengan algoritma yang berbeda yang menggunakan kunci *public* seperti RSA.
- Ukuran *file* yang dihasilkan dapat diperkecil dengan menerapkan proses kompresi data.

DAFTAR PUSTAKA

[1] Adriansyah, Yusuf. 2010. Enkripsi Sederhana dengan *Base64* dan Subtitusi Monoalfabetik ke Huruf Non-Latin. Makalah Mahasiswa Teknologi Bandung.

[2] Ariyus, Dony., 2008 Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi, Andi Offset,Yogyakarta.

[3] Hendarsyah, Decky& Wardoyo,Retantyo (2012) Implementasi Protokol *Diffie - Hellman* Dan Algoritma RC4 Untuk Keamanan Pesan SMS. Universitas Gajah Mada

[4] Kurniawan, J., 2004, Kriptografi, Keamanan Internet dan Jaringan Komunikasi, Informatika Bandung.

[5] Schneier, B., 1996, *Applied Crytography : Protocols, Algorithm, and Source Code in C*, *Second Edition*, John Willey and Sons Inc.

[6] Thorsteinson, Peter dan G. Gnana Arun Ganesh, 2004, *.NET SECURITY and CRYPTOGRAPHY*.USA : Pearson Education, Inc

[7] Borenstein, N. & N. Freed, 1996. *Multipurpose Internet Mail Extensions (MIME) Part One : Format of Internet Message Bodies*. RFC 2045. *Network Working Group*

[8] Suriski Sitinjak, Fauziah, Yuli Juwairiah, "APLIKASI KRIPTOGRAFI FILE MENGGUNAKAN BLOWFISH", Seminar Nasional Informatika, 2010

[9] Ahmad Timbul Sholeh, Erwin Gunadhi, Asep Deddy Supriatna, "MENGAMANKAN SKRIP PADA BAHASA PEMOGRAMAN PHP DENGAN MENGGUNAKAN KRIPTOGRAFI BASE64", 2013

[10] Reza Fitra Kesuma "PEMBUATAN PERANGKAT LUNAK SEBAGAI MEDIA PEMBELAJARAN KRIPTOGRAFI MODERN METODE BLOWFISH", Naskah Publikas, 2013

[11] Andy Nugroho "IMPLEMENTASI ALGORIMA CAESAR CIPHER ROT13 DAN BASE64 UNTUK ENKRIPSI DAN DESKRIPSI PESAN SMS PADA HANDPHONE BERBASIS ANDROID", Naskah Publikasi, 2012

[12] Ari Suhendra "ANALISIS DAN IMPLEMENTASI ENKRIPSI BASIS DATA DENGAN ALGORITMA KRIPTOGRAFI BLOWFISH", Naskah Publikasi, 2012

[13] Shanty Erikawaty Tambunan "IMPLEMENTASI ALGORITMA KRIPTOGRAFI BLOWFISH UNTUK KEAMANAN DOKUMEN PADA MICROSOFT OFFICE", Naskah Publikasi, 2010

[14] http://sinta.ukdw.ac.id , IMPLEMENTASI ALGORITMA KRIPTOGRAFI BLOWFISH UNTUK ENKRIPSI-DEKRIPSI CITRA DIGITAL KE DALAM BENTUK TEKS, di akses tanggal 19 Juni 2015